Algorithms, Data Structures & Complexity

Lab 2: Sorting Algorithms

Due on first session of lab 3 for your group

Federico Pecora, Uwe Köckemann

Uwe Köckemann

Handing In

This lab should be completed and shown during the first session of lab 3 for your group. The TA will pass by your seat and evaluate each exercise. Upon successful completion of the lab, for each lab exercise, please provide a text file named ex_n.txt with the following content:

- indicate which file(s) implement the algorithm and/or data structure in the exercise;
- a brief explanation of the tests that were carried out to test the implementation;
- instructions on how to execute a test to verify the implemented code;
- answers to any theoretical questions asked in the exercise.

Please submit all lab material collected into an archive (zip, rar, or tar.gz) via a Blackboard message to Uwe Köckemann and Federico Pecora.

Note: labs should be done in pairs. Larger groups are *not* allowed. All incidents of plagiarism will be reported. Please write your names on all material you hand in.

Exercise 1 — Sorting I

Implement a sorting algorithm with quadratic complexity.

Exercise 2 — Sorting II

Implement a sorting algorithm with better than quadratic complexity.

Exercise 3 — Asymptotic Complexity of Sorting Algorithms

Choose the best algorithm (among Insertion Sort, Merge Sort and Quick Sort) to use in the following cases and explain your decision:

- The input array is sorted in descending order;
- The input array is sorted in ascending order;
- All elements in the array are the same.
- There are both positive and negative numbers in the array.

Exercise 4 — Testing I

Test your implementations with

- the lists given in Lab 1;
- your Swedish personal number (personnummer).

Exercise 5 — Testing (II)

Test the program on the sorting problems provided with this lab:

1. Use the *load_file* function to load one of the provided files (see below) into an array
2. Sort the array using the algorithms implemented in this lab
3. Print all numbers in sorted order

4. How do algorithms with quadratic complexity compare to better ones on problems of different sizes?

- The file *load_files.tar.gz* contains a small library to load files into an array
- The file *sorting_problems.tar.gz* contains files with random numbers between 10 and 100000 (and a python script to generate more problems)